DoD SOA Symposium 2010

Preparing for Semantic Technology in SOA

Paul Keller

NASA Ames Research Center

April 21-22, 2010

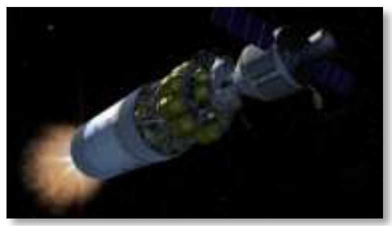
Background

- NASA undertakes multi-decadal programs in science and engineering
- The Agency is involved in all aspects of a program, from conceptualization, design, manufacture, test, operations, through retirement
- IT systems and data need to support a large, distributed, and evolving stakeholder community and technology base

Challenge for Constellation Program

- NASA's Constellation Program is implementing the Vision for Space Exploration (VSE), operating a series of coordinated human and robotic missions, including surface systems, over a period of 30+ years

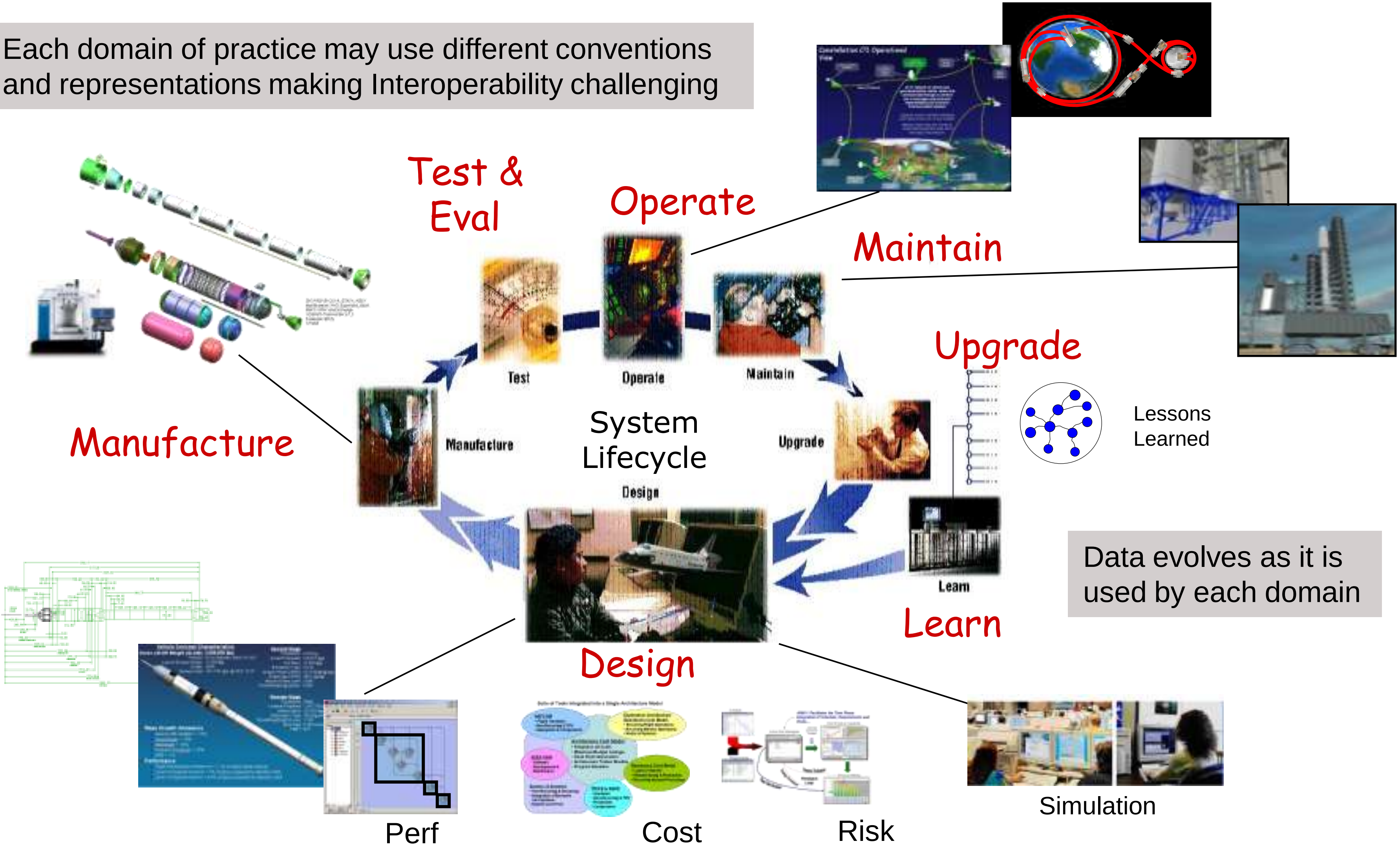




- How to effectively support this long-term effort given the expected technology advances, varying stakeholder needs, and allow for planned and unplanned evolution/changes?

Supporting the Lifecycles

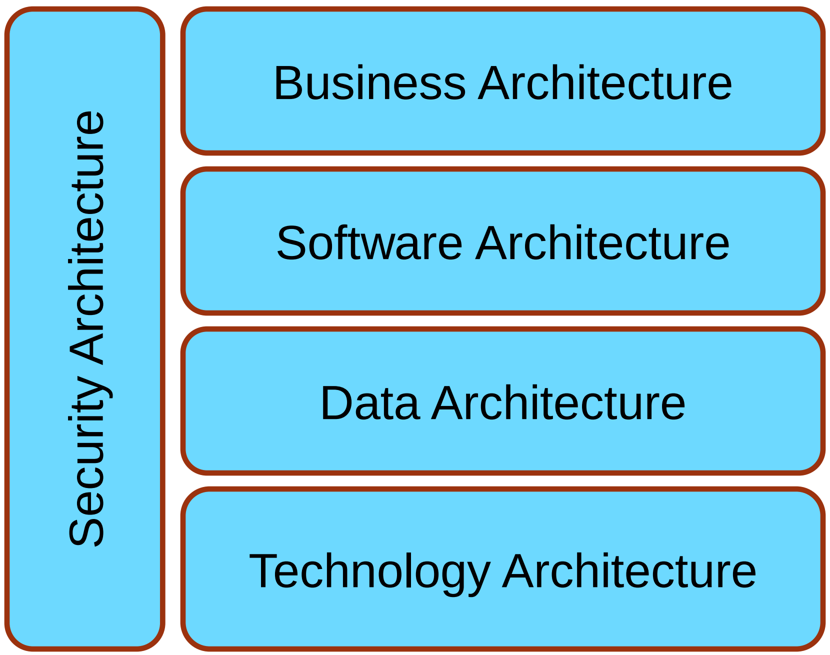
Each domain of practice may use different conventions and representations making Interoperability challenging



Data evolves as it is used by each domain

Selecting an Enterprise Architecture

- Examined many of the current EA’s and found that a nominal TOGAF architecture aligned with the organization, missions, systems



The challenge for most organizations is to truly adhere to an EA

For example, is there is a clear separation between your data and software?

General Needs

- Ability to use distributed systems – many partners, vendors, contractors, etc. (producers and consumers)
- Need evolvable, flexible data representation
- Need evolvable, flexible IT systems
- But…
  - It is not reasonable to rewrite or replace or even abstract every system
  - Solution needs to have adoption scaling (“buy by the yard”)

An Information Model Continuum

Information models are pervasive, but there is a wide range of representations

Pump: “a device for moving a gas or liquid from one place or container to another”

Shared human consensus	Text descriptions	Information hardwired; used at runtime	Information processed and used at runtime
Implicit	**Informal** (explicit)	**Formal** (for humans)	**Formal** (for machines)

Further to the right means:

- Less ambiguity
- More likely to have correct functionality
- Better interoperation
- Less hardwiring
- More robust to change
- More effort – needs up front investment in understanding the information

Source: “Where are the Semantics in the Semantic Web”, Michael Uschold, The Boeing Company

Cx Data Architecture Objectives

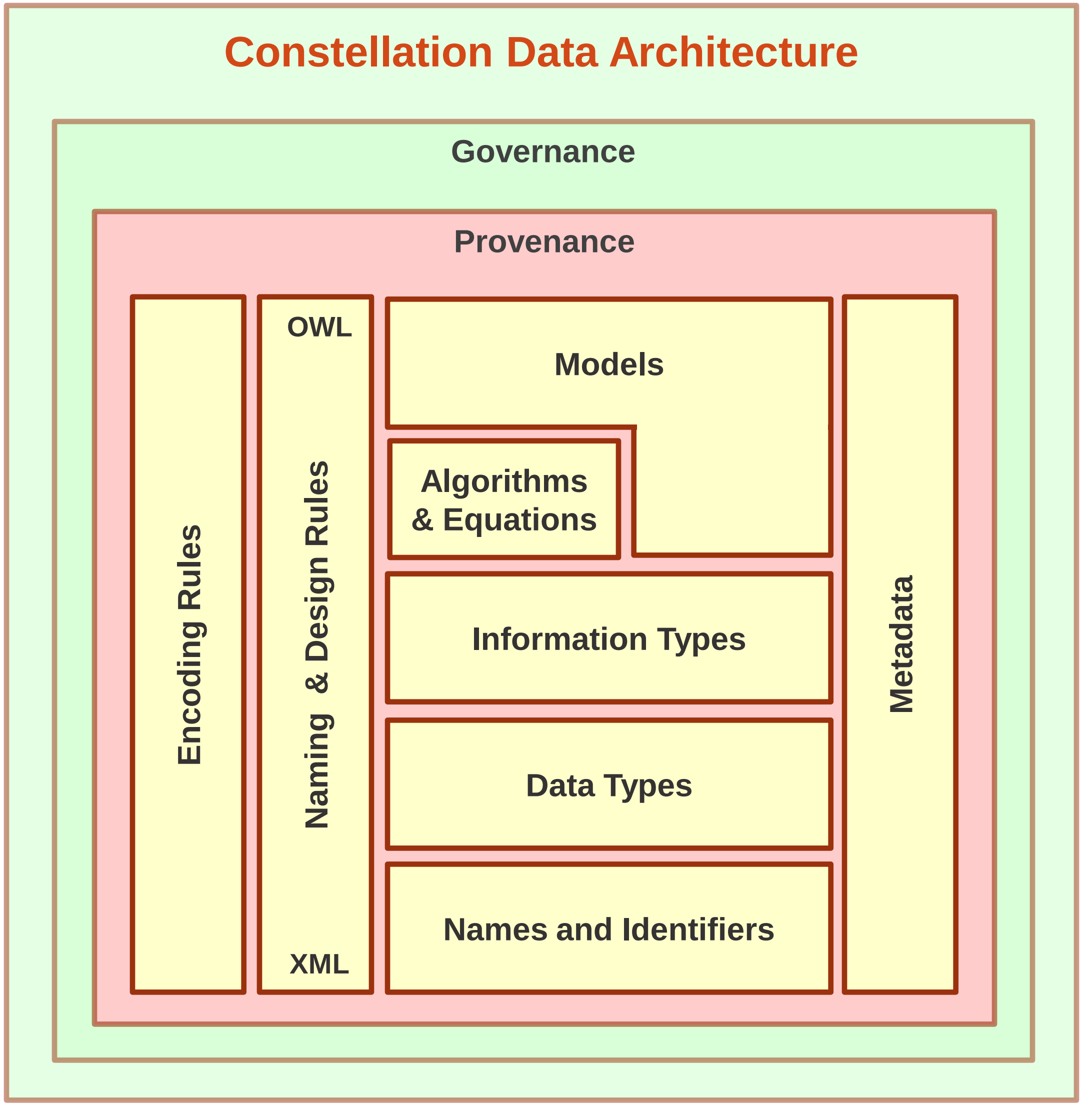
1. Semantic Interoperability - support interoperability between Constellation Elements through controlled information types, structures, and knowledge of their meanings, and specified abstraction and composition rules
2. Semantic Mediation - specify structures for, and mediate exchange and aggregation of, information between and among tools, systems and users
3. Knowledge Capture - capture knowledge across mission and system lifecycles, including long-term maintenance and evolution
4. Knowledge Flow - define and deploy models of mission and system lifecycles for unifying knowledge capture and flow across NASA disciplines and teams
5. Information Exchange - provision operational information exchange schemas for specific needs with guarantees of consistency and completeness of representation
6. Information Specification - define information structures and naming within ESMD programs with precision, traceable provenance, and governance
7. Information Independence - represent information in a tool-neutral and project independent manner
8. Operational Effectiveness - improve decision support, throughout the lifecycle, for risk, performance, and costs starting with trade studies and through complete mission and system lifecycles, including long-term maintenance and evolution

These objectives drove our decision to use Semantic Technologies as the basis for the CxDA Data Architecture

Approach for Data and IT Architecture

- Semantic Technology –based approach for Data Architecture
  - Basis for application-independent neutral model for interoperability
  - Machine-intelligable representations
  - Provides a model-based data specification
- SOA approach for IT Architecture
  - Distributed Systems, Loose coupling, Flexible Service Composition, Composable Data Flows, Abstraction
  - Semantic technology –enabled Registries

Constellation Data Architecture (CxDA)



Using the CxDA

- Information Architecture is the practice of *structuring information* for a purpose
  - Establish authoritative source of information interoperability and data architecture standards
  - Use tool-independent representations for interoperability
  - Capture relationships between information
- High Level Elements

Need	Element	Description
What to call it?	Identification	Nomenclature, Identifiers, & Terminology
How to represent it?	Data Exchange	Common Formats & Protocols
How to describe it?	Models	Formal Descriptions of Information
How to implement it?	Infrastructure	Services & Registries
How to use it?	Process	Data Flows & Management
Where to get it?	Data Assets	Original Sources & Data Repositories

CxDA/NExIOM Standard Vocabulary (NSV)

- Basic physical quantities, forces & moments examples

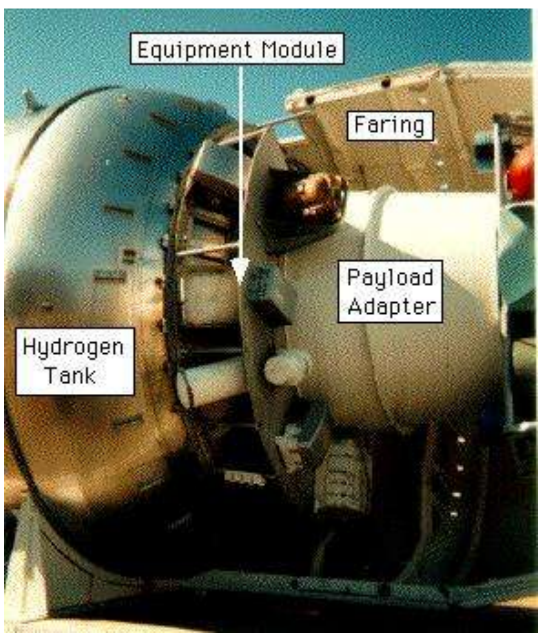
Data-Name Identifier	Description	Definition	Symbol (Units)	Units
Potential	Potential	$\nabla\varphi = q$	L^2/T	SI
StreamFunction	Stream function (2-D)	$\nabla \times \psi = q$	L^2/T	SI
Density	Static density	(ρ)	M/L^3	SI
Pressure	Static pressure	(p)	$M/(LT^2)$	SI
Temperature	Static temperature	(T)	$\boldsymbol{\Theta}$	SI
EnergyInternal	Static internal energy per unit mass	(e)	L^2/T^2	SI
Enthalpy	Static enthalpy per unit mass	(h)	L^2/T^2	SI
Entropy	Entropy	(s)	$ML^2/(T^2\Theta)$	SI
EntropyApprox	Approximate entropy	$(s_{app} = p / \rho^\gamma)$	$(L^{(3\gamma-1)})/((M^{(\gamma-1)}).T^2)$	SI
DensityStagnation	Stagnation density	(ρ_0)	M/L^3	SI
PressureStagnation	Stagnation pressure	(p_0)	$M/(LT^2)$	SI
TemperatureStagnation	Stagnation temperature	(T_0)	$\boldsymbol{\Theta}$	SI
EnergyStagnation	Stagnation energy per unit mass	(e_0)	L^2/T^2	SI
EnthalpyStagnation	Stagnation enthalpy per unit mass	(h_0)	L^2/T^2	SI
EnergyStagnationDensity	Stagnation energy per unit volume	(ρe_0)	$M/(LT^2)$	SI
VelocityX	x-component of velocity	$(u = q \cdot \mathbf{e}_x)$	L/T	SI
VelocityY	y-component of velocity	$(v = q \,.\, \mathbf{e}_y)$	L/T	SI
VelocityZ	z-component of velocity	$(w = q \,.\, \mathbf{e}_z)$	L/T	SI
VelocityR	Radial velocity component	$(q \,.\, \mathbf{e}_r)$	L/T	SI

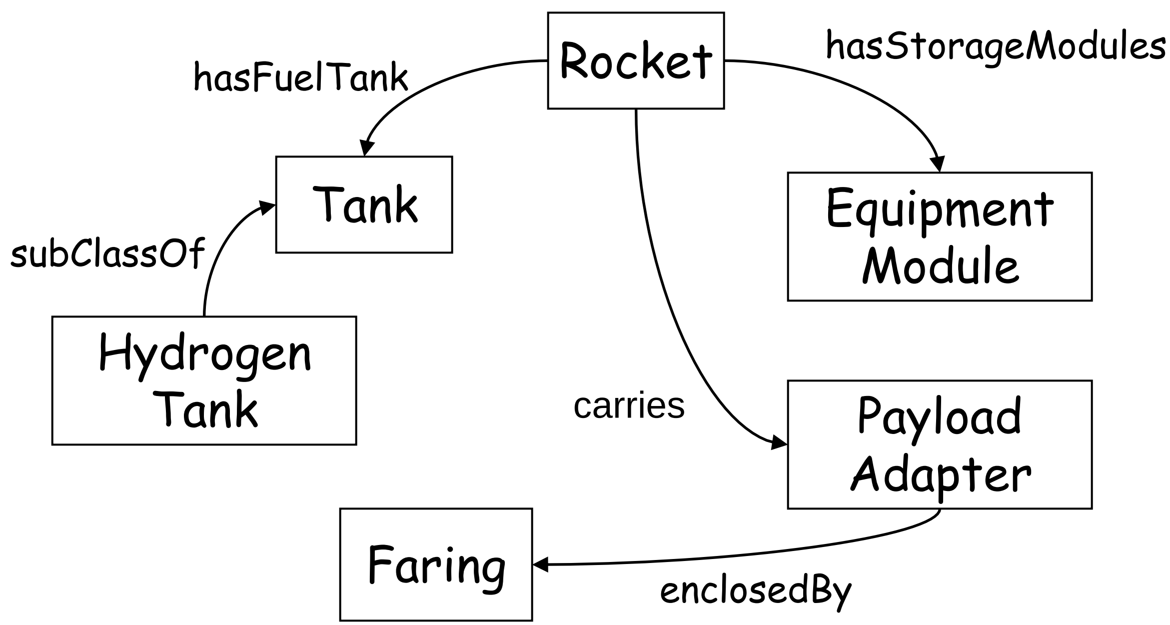
Data-Name Identifier	Description	Units
ForceX	$F_x = \mathbf{F} \cdot \mathbf{e}_x$	**ML/T^2**
ForceY	$F_y = \mathbf{F} \cdot \mathbf{e}_y$	**ML/T^2**
ForceZ	$F_z = \mathbf{F} \cdot \mathbf{e}_z$	**ML/T^2**
ForceR	$F_r = \mathbf{F} \cdot \mathbf{e}_r$	**ML/T^2**
ForceTheta	$F_\theta = \mathbf{F} \cdot \mathbf{e}_\theta$	**ML/T^2**
ForcePhi	$F_\varphi = \mathbf{F} \cdot \mathbf{e}_\varphi$	**ML/T^2**
Lift	L or L'	**ML/T^2**
Drag	D or D'	**ML/T^2**
MomentX	$M_x = \mathbf{M} \cdot \mathbf{e}_x$	**ML^2/T**
MomentY	$M_y = \mathbf{M} \cdot \mathbf{e}_y$	**ML^2/T**
MomentZ	$M_z = \mathbf{M} \cdot \mathbf{e}_z$	**ML^2/T**
MomentR	$M_r = \mathbf{M} \cdot \mathbf{e}_r$	**ML^2/T**
MomentTheta	$M_\theta = \mathbf{M} \cdot \mathbf{e}_\theta$	**ML^2/T**
MomentPhi	$M_\varphi = \mathbf{M} \cdot \mathbf{e}_\varphi$	**ML^2/T**
MomentXi	$M_\xi = \mathbf{M} \cdot \mathbf{e}_\xi$	**ML^2/T**
MomentEta	$M_\eta = \mathbf{M} \cdot \mathbf{e}_\eta$	**ML^2/T**
MomentZeta	$M_\zeta = \mathbf{r} \cdot \mathbf{e}_\zeta$	**ML^2/T**
Moment_CenterX	$x_0 = \mathbf{r}_0 \cdot \mathbf{e}_x$	**L**
Moment_CenterY	$y_0 = \mathbf{r}_0 \cdot \mathbf{e}_y$	**L**
Moment_CenterZ	$z_0 = \mathbf{r}_0 \cdot \mathbf{e}_z$	**L**

More at http://www.qudt.org/

CxDA Ontology Modeling

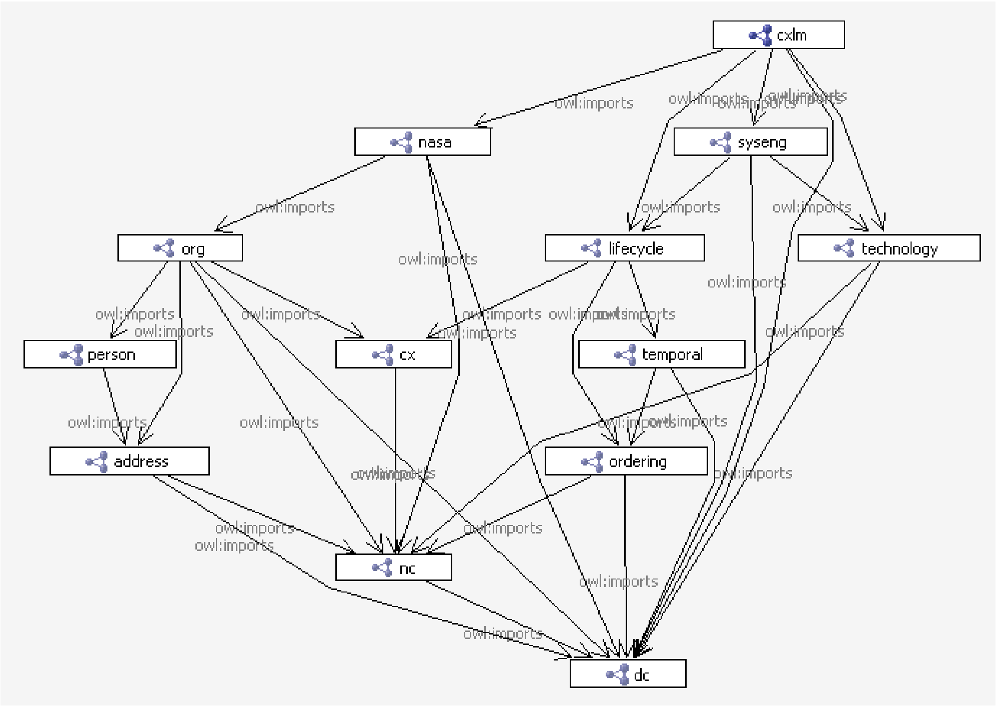
- Concepts and relationships used to describe and represent an area of knowledge (subject matter) are defined
  - A vocabulary
  - A shared language
  - An explicit representation of relationships
- Used by people, databases, and applications that need to share information in a subject area
- A variety of representations are available, both human and machine readable





Ontology Architecture

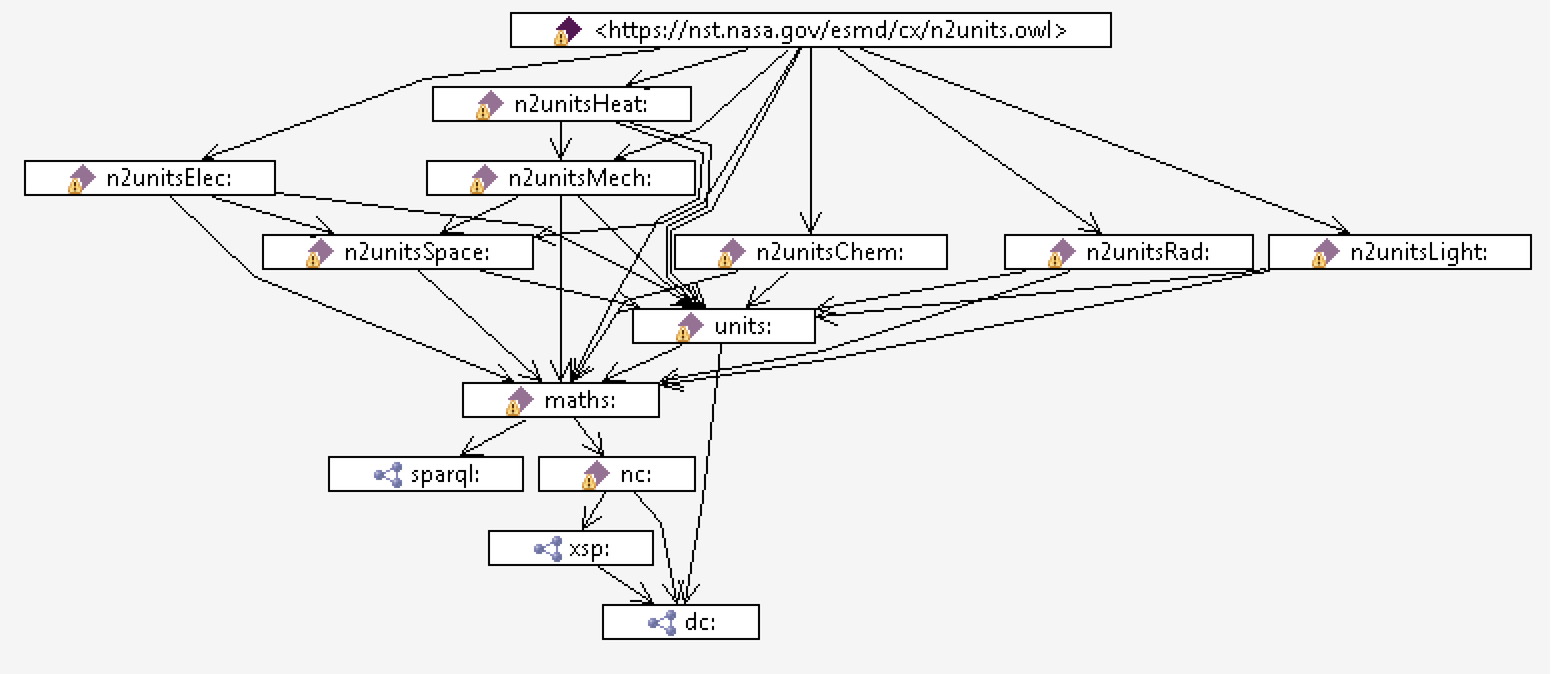
- Ontology Architecture developed to support componentization, reuse, composition, and co-development
- Integrates domains and disciplines
- Over 200 ontologies (and growing)
- Substantial work on fundamental models for engr/physics/ops



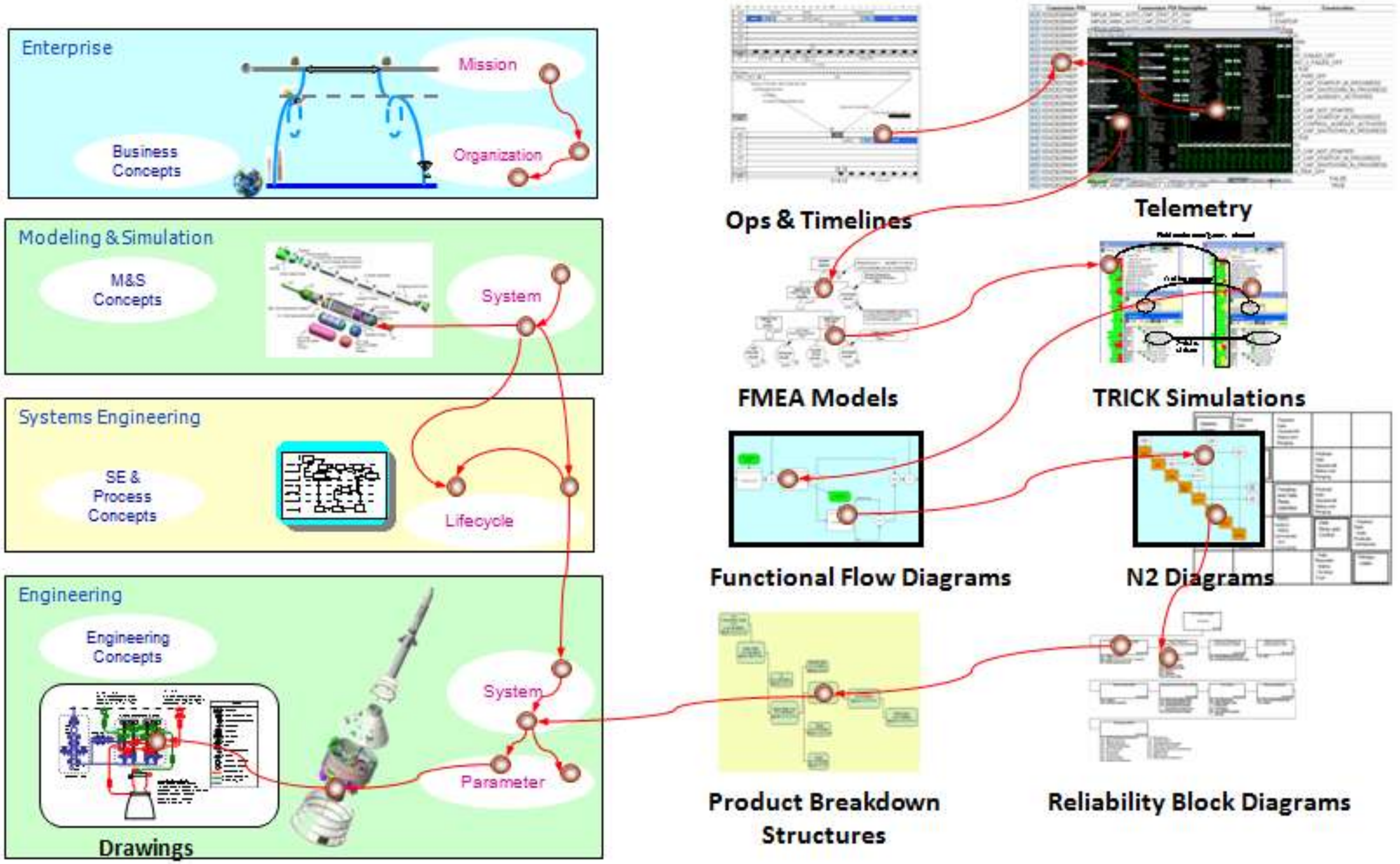
Ex: Ontology Architecture of the Constellation Lifecycle Model

Units Ontology

- The Units Ontology is composed of several individual ontologies



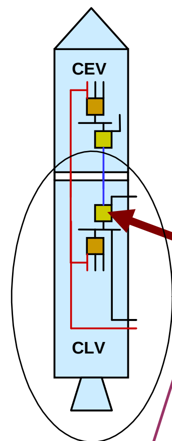
Connecting your Information

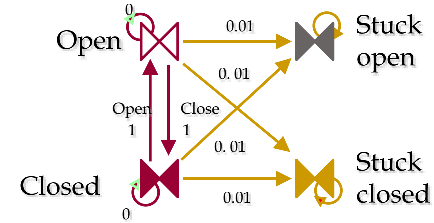


Putting the Models to Work

CxDA Models Allow Traceability from Constellation Element to Component, Parameters, Values and Units

Constellation System: CLV

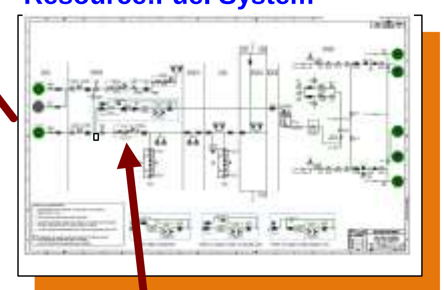




Data Exchange

Sub-System: Propulsion

Resource:Fuel System

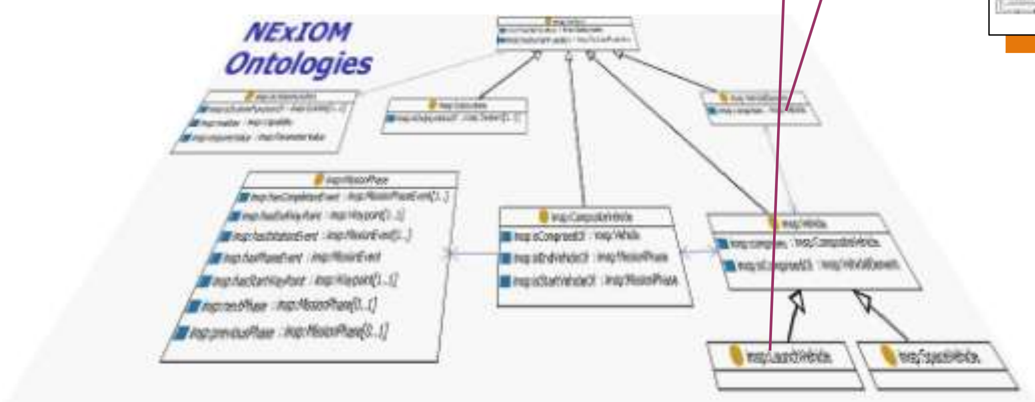


Specify Data and Data Structures

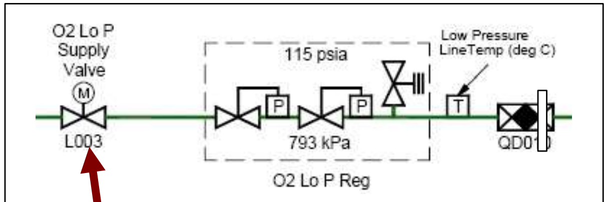
Data Discovery

Processing Data

Associate Data to processes, SW, HW, orgs



Device: O2 Lo P Supply Valve



Parameter: Valve State

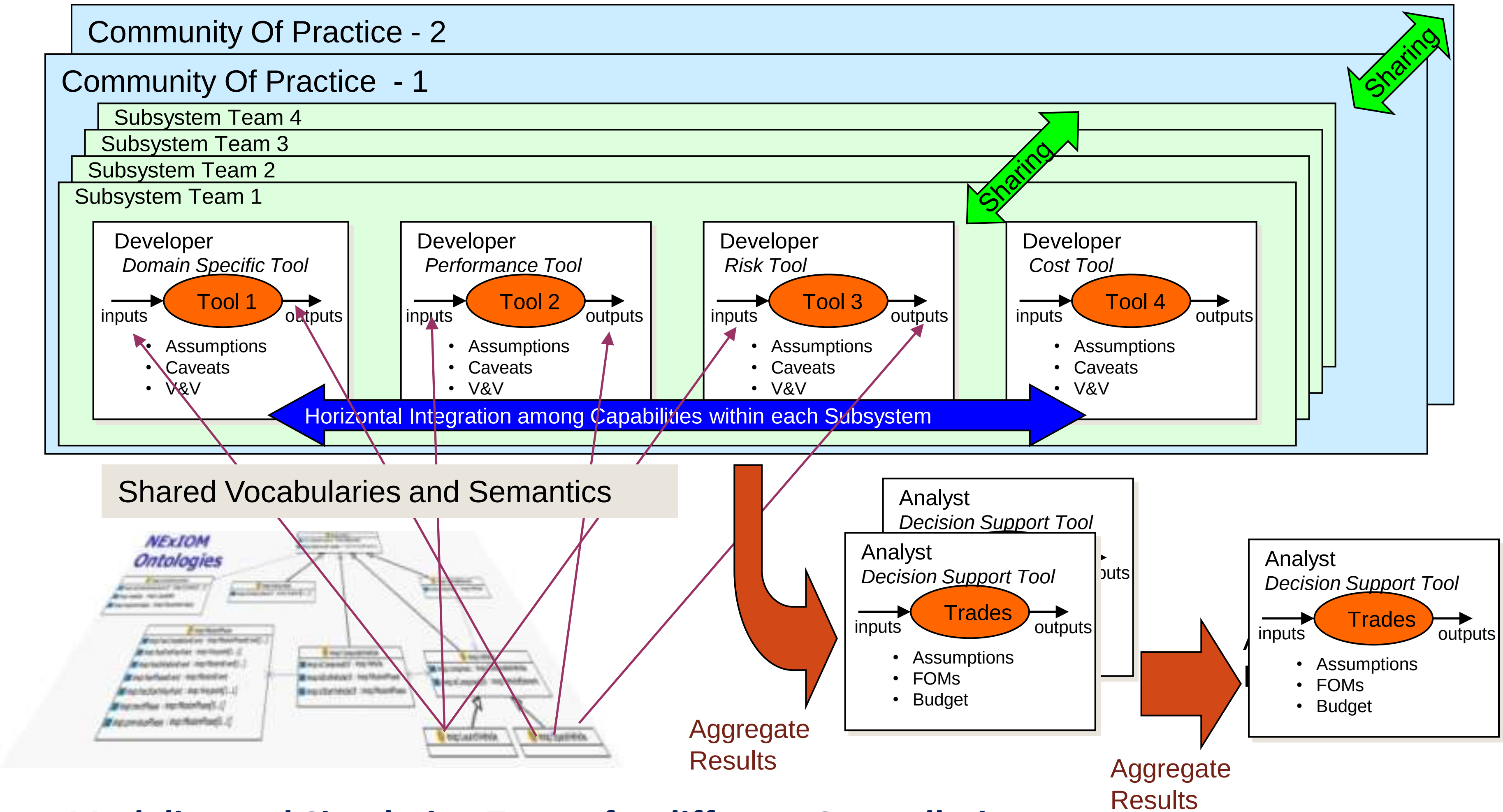
Parameter	Type	Value	Units
Flow Rate	Real	258.75	l/sec
Valve State	ValvePosn	Open	None

Understand Pedigree of Data

Define Relationships between Data

Understand Authority of Data

SOA in Design and Analysis Tool Chains



Modeling and Simulation Teams for different Constellation Elements use tools that need to interoperate

Is XML enough?

- XML was originally defined to describe documents
  - Effectively only one structuring tool
    - Tree/Hierarchy/Container
  - Weak support for relationships
  - Weak support for merging/combining data (aggregation)
  - No consistent method of defining semantics
  - Schema limitations

- XML provides standardization of data format and processing tooling, but not data modeling
  - XML does equal Interoperability

Looking at OWL to help

- OWL is based on set theory = Graphs not Trees
- Far more expressivity as a data modeling language than XML
- Strong type system
- Strong support for relationships (first class objects)
- Foundational specification of identity/addressability
- Schema uses the same language as Data spec
  - “levels the playing field” between Schema and Data
  - Other work provides Rules (SPIN) using the same representation as Schema and Data

Considerations for Data Exchange

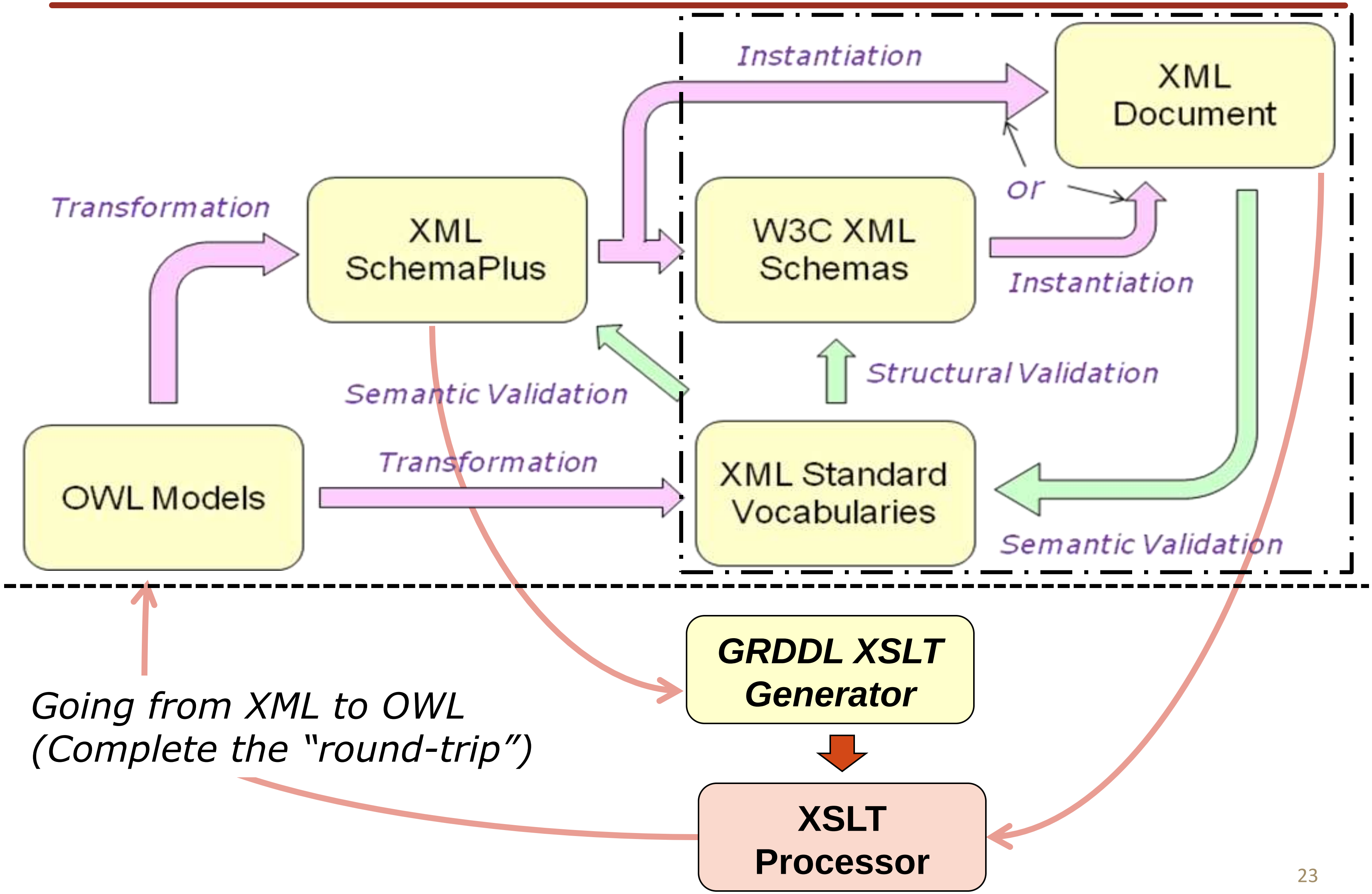
- Ideally would exchange data using (ontology) models (RDF/OWL)
  - But semantic technology is a relatively new practice

- So how to accommodate reality of the existing and growing XML ecology?
  - XML SchemaPlus
  - Fully XML compliant, but retains ontology basis
  - Fundamental coexistence strategy

XML SchemaPlus

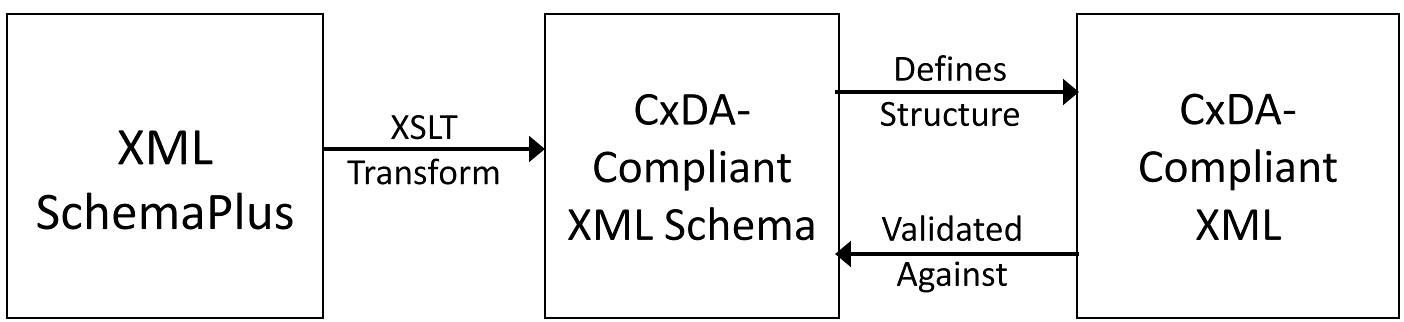
- XML SchemaPlus (XSP) was developed to integrate Semantic Models (RDF/OWL) with XML
- XML has numerous challenges as a data modeling language
- XML SchemaPlus introduces specification and methods of specifying XML that resolves model issues
  - Result is XML Schema and Data files that are fully compliant with XML standard, but align with models in Ontologies
  - XSP XML is much cleaner, simpler, powerful, and precise
  - Using controlled vocabularies and available structuring mechanisms XML can be authored and consumed without direct use of ontologies
  - More at http://www.xspl.us/

Application XML Schemas and NASA Controlled Vocabularies are generated from OWL Models



Generating XML Schemas from SchemaPlus

- A SchemaPlus can be converted to an XML Schema document (XSD) automatically
  - The transform SchemaPlusToSchema.xslt is used to generate the schema from the SchemaPlus
- SchemaPlusToSchema.xslt is part of the NExIOM XML SchemaPlus suite
  - If you develop your own SchemaPlus, you can run this transform to generate an XSD from it



Service Implementation

- Wanted to have an architecture which strongly supported ability to easily "stand up systems"
- In the Constellation Program not all applications lend themselves to providing a service interface and participation in a SOA
  - Legacy applications, insufficient APIs, insufficient performance, access control restrictions
- The use of Registries was introduced
  - Takes on a variety of participation roles including assist, augment, replace, new capability

Using Semantics with Registries

- Using ontologies allow a data model to be componentized, with clear distinctions, constraints, and relationships established
  - An ontology architecture defines how the componentized models are used in concert
- Using semantic technology –enabled Registries can be thought of as an implementation of the CxDA ontologies
  - can use multiple registries to split up data in meaningful ways
  - A SOA architecture allows registries to be used in concert

System of Registries

- Provide consistent specifications of data
  - Across time, between organizations, between processes
  - With explicit specification of machine intelligable semantics
- Connect "silos" of information
  - Captured within applications or proprietary file formats, through the use of standardized data definitions
- Support the exchange of information
  - Using consistent formats and protocols; ontology-compliant XML and Web Services

CxDA Registries in SOA

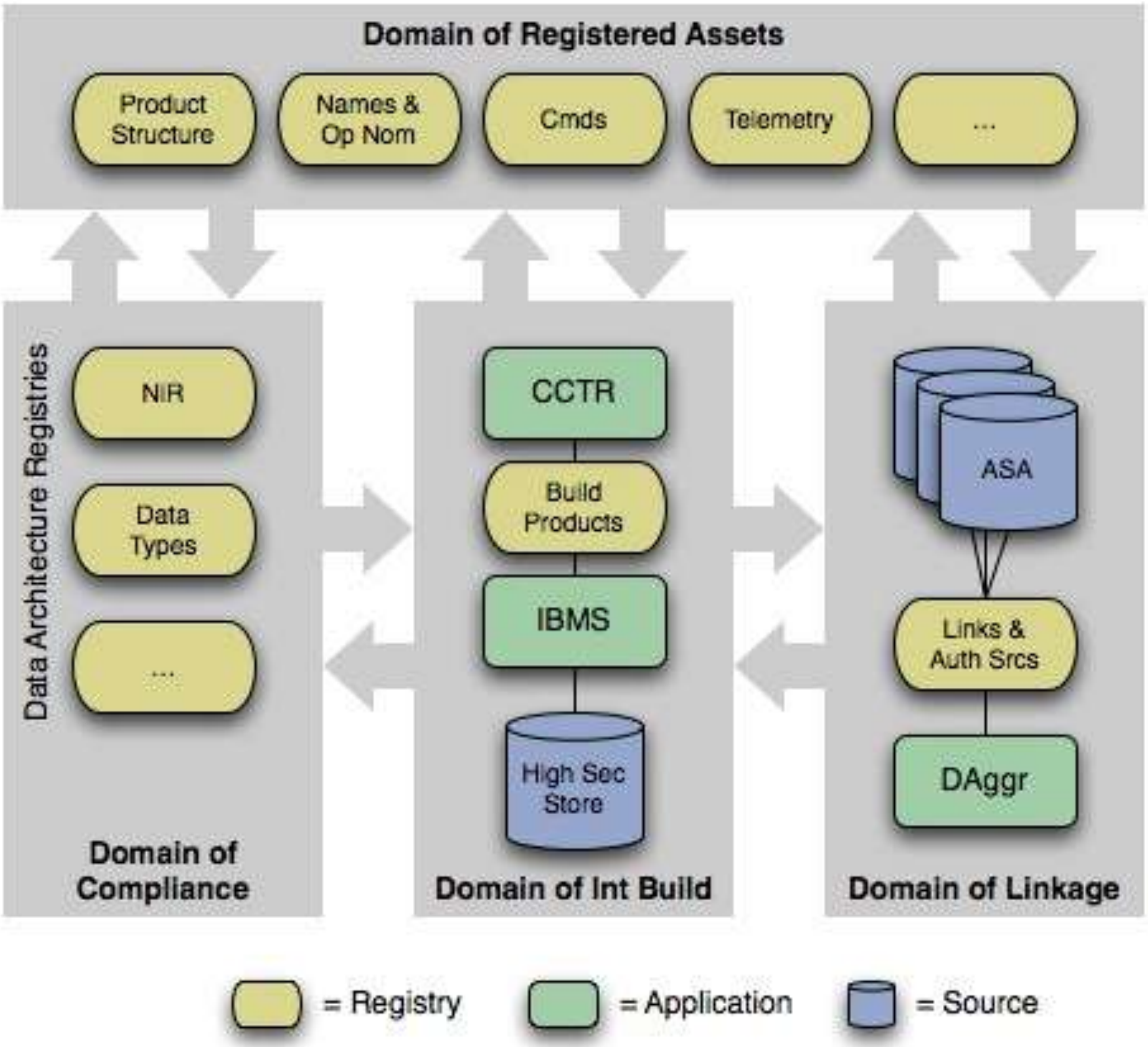
- Use registries with native capabilities of semantic web triplestores
  - URI's, distributed deployment, query, aggregation
- Registries serve many functions in the architecture
  - Definitions
  - Check/Audit (Syntax, Structural, Semantic)
  - Locating information – incl. search & query
  - Connecting information
  - Mediation, incl. Transformation
  - Application abstraction
  - Data (!)
  - …

Example use of Registries and Applications

This diagram is a high-level depiction of a registry-enabled SOA supporting missions operations

The applications (IBMS, CCTR, DAggr) can interact with and can be based on a variety of registries to support functions in the various domains.

ASA = Authoritative Source Application



Conclusion

- Semantic Technology approach provides key capability for a flexible SOA
- Model-based data definitions (Data Architecture) are essential for interoperability and data longevity
  - Commit more to models, less to code
- An ontology-compatible approach to representing information in XML format has been developed to enable usage in existing applications
- Use of registries can provide implementation flexibility and functionality in SOA’s